DOCUMENT RESUME

ED 061 770 EM 009 831

AUTHOR Shapiro, Peter D.
TITLE After Data Collection: Coding-An Educational Research Tool.
INSTITUTION Stanford Univ., Calif. ERIC Clearinghouse on Educational Media and Technology.
SPONS AGENCY Office of Education (DHEW), Washington, D.C.
PUB DATE Apr 72
CONTRACT OEC-1-7-070-873-4581
NOTE 14p.

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Codification; Reliability; *Research Methodology; Research Problems; *Statistical Analysis; Validity

ABSTRACT

A brief and simple guide discusses the place and purpose of coding experimental data in the research process. The trade-offs involved unitizing data are reviewed; it is noted that a decision that increases reliability may reduce the validity of results, but without reliability there will be no validity at all. A discussion of categorizing data includes information on building the category set, the relationship of categories within a set, and designing the discrete-category code. Ways of calculating reliability are presented to allow the researcher to measure the equivalence of results when different coders classify the same data using the same set of categories. (JY)

ED 061770

An ERIC Paper

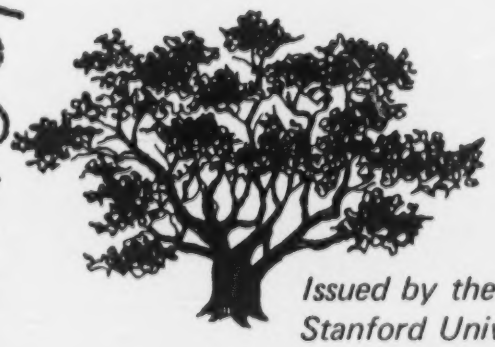
AFTER DATA COLLECTION: CODING—AN EDUCATIONAL RESEARCH TOOL

By Peter D. Shapiro

Institute for Communication Research
Stanford University
Stanford, California

April 1972

EM 009 831



Issued by the ERIC Clearinghouse on Media and Technology
Stanford University, Stanford, California 94305

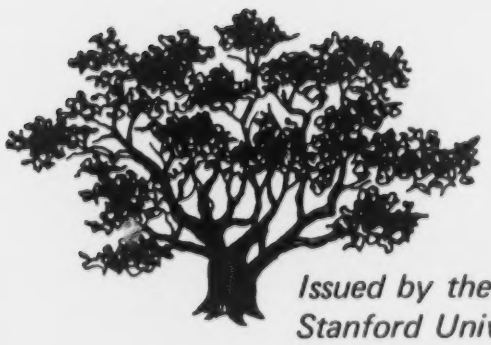
U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

AFTER DATA COLLECTION: CODING—AN EDUCATIONAL RESEARCH TOOL

By Peter D. Shapiro

Institute for Communication Research
Stanford University
Stanford, California

April 1972



Issued by the ERIC Clearinghouse on Media and Technology
Stanford University, Stanford, California 94305

ACKNOWLEDGMENTS

Thanks are due to William J. Paisley, of the Institute for Communication Research, whose insights and guidance made possible the writing of this paper. Don H. Coombs, Director of the ERIC Clearinghouse on Media and Technology, and Katherine Miller also contributed valuable suggestions.

TABLE OF CONTENTS

I.	Coding: Its Place and Purpose	1
	Who Does the Coding?	1
II.	Unitizing the Data	1
III.	Categorizing the Data	2
	Building the Category Set	3
	Relationship of Categories Within a Set	3
	Designing the Discrete-Category Code	4
IV.	Calculating Reliability	4
	Formulas	5
	A. Correlation	5
	B. Agreement Statistics	6
V.	Conclusion	8
References		9

I. CODING: ITS PLACE AND PURPOSE

Your interviewers have piled your desk with survey questionnaires that are filled with long, complex responses. Or you are analyzing sex and race stereotypes and have collected samples of textbook content. Or you possess videotape records of classroom interactions under different conditions, and you want to compare them.

In these cases, and in others like them, you must now do some coding, since coding is the research task that intervenes between data collection and analysis. What follows is a simple and practical guide to help you through the decision steps that mark the coding process.

Very simply, you code by (1) dividing your material into units, (2) designing categories that reflect the research questions, (3) comparing the units with the categories and placing each unit in the appropriate category, and (4) counting the number of units classified under each category. Then you have a profile of your data comprised of frequencies of classifications for each category. The frequency profile may be sufficient as a test of your hypothesis or as an answer to your research question, or you may want to compare it with other quantitative data in order to reach conclusions.

For instance, in a study of sex stereotypes in textbooks, you might compare the proportion of women characters employed as "professionals" with the proportion of men characters so employed. Further, the adjectives used to describe male and female professionals could be coded and analyzed.

The primary criterion of success in coding is *reliability*, which will be discussed in detail later. Briefly, if the coding is reliable, you can say that the pattern or profile that has been abstracted from the data has been abstracted objectively. If another person coded the same data using the same procedures, that is, he would get the same results. If the coding is reliable, your research can be replicated. Unreliable coding, perhaps based on mistaken interpretation of the categories, makes it impossible to have confidence in your findings.

Who Does the Coding?

Coding is an important but generally unexciting job. If you have the money, you will probably hire others to do the bulk of it.

Doing It Yourself. If you do the coding yourself, you should attempt to establish a common interpretation of your categories with another person. Agreement between you and the other person could be tested on a sample of the units to be coded. Working with another person reduces the likelihood of idiosyncratic coding, and allows you to legitimize your results with a reliability statistic.

Hiring Several Coders. When you have several coders, you should establish in them a *common frame of reference* with respect to your categories. The coders should all be in a state of readiness to actually perceive the data in terms of the categories and to ignore extraneous cues. Coding, after all, is nothing more than the practice of controlled and applied perception.

In training its coders, the Survey Research Center at the University of Michigan (SRC, 1965) first has them discuss the categories together and reach a group interpretation that is consistent with the project director's interpretation. Then each coder codes a sample of questionnaires to bring problems to light. In further discussion, these are ironed out. Coders who cannot adopt the common frame of reference have to be either re-trained or dismissed.

In later stages of coding, SRC employs "check-coding" to monitor the reliability of individual coders, to keep coders within the frame of reference, and to identify problems created by unclear categories. A percentage of each coder's questionnaires are re-coded by a check-coder (another coder, the supervisor, or a member of the analysis staff). Differences between coders and check-coders are recorded, and disagreements settled in discussion. The percentage of questionnaires check-coded is likely to be larger near the beginning of coding and then to level out. This percentage, which varies from 10 to 30% of the questionnaires at SRC, is also larger if the code has been giving difficulties or if the findings of the study are especially important.

Hiring a Computer. Sometimes it is possible to hire a computer to do your coding. Computers are best for large scale repetitious counting tasks and poorest when ability to judge between difficult coding choices is required.

II. UNITIZING THE DATA

The *unit* is the smallest division or segment of content that is to receive a score. You may decide to code only simple units such as words or symbols. For example, in comparing textbook descriptions of male and female professionals, you might code physical-appearance and competency adjectives. Or you

may wish to code complex units such as the themes that underlie the words. Like most decisions, the unitization decision involves a trade-off between different values. How to unitize your data must be determined in the light of your research goals and your resources.

Simple Units. Deciding to use simple units permits a high degree of coding efficiency and reliability. The simple unit has clear syntactic boundaries. You have only to locate the unit in order to classify it and coding then becomes an easy counting task.

In fact, computers can code simple units with perfect reliability. A popular set of computer procedures, the General Inquirer, can process natural text and locate, count and tabulate text characteristics (Stone, 1969; Goldhamer, 1969). The General Inquirer now includes a family of dictionaries, data preparation systems and analysis programs that are being used in all the social sciences. Each word in a General Inquirer dictionary is characterized by denotative and connotative tags that are relevant to the research. For example, Holsti (1969) reports research in which he tagged words along three dimensions: positive-negative, strong-weak and active-passive. Words in the dictionary also were assigned intensity ratings, from 1 to 3, along each of the dimensions. Thus a particular word such as "accost" was tagged in the dictionary as follows: negative (3), strong (2), active (2). Holsti's dictionary was used in analyzing the statements of foreign policy decision-makers.

This is how the General Inquirer works: You enter the text to be analyzed into the computer on cards or through typewriter terminals. The contexts of the words can be taken into account as they are matched with their counterparts in the dictionary. Then the words are counted and weights are assigned to the tendencies that interest the researchers. For example, John Foster Dulles' beliefs about the Soviet Union were characterized in terms of his negative affect towards that country, and his image of its strength and activeness (cited in Holsti, 1969).

Computer coding requires special preparation of the text. Thus the computer is economical only under certain conditions: (a) when there is a large amount of data including very frequent occurrence of simple units, or (b) when the text will be re-analyzed by different researchers, or (c) when the data will be used in continuing studies, or (d) when the data are already available in machine-readable form.

The Trade-Off. While your choice to code simple units permits high reliability (up to 100% if you use a computer), you may at the same time be sacrificing external validity. How much can word frequencies tell us about our research questions? All social research that bases its conclusions on frequencies of events is subject to critical appraisal, but research relying on the coding of simple units is especially vulnerable. For instance, Holsti (1969) was skeptical about one researcher's assumption that frequency of place names in a sample of newspapers could be taken as an index of community awareness.

Complex Units. Deciding to code complex units may make your results more valid in that complex units are probably better than simple units as operational definitions of your concepts. For example, say you are exploring sex stereotyping in textbooks: your results will be closer to what you mean by "stereotyping" if you code themes such as "professional women are unattractive" than if you just count the number of unflattering adjectives that depict such women.

With complex units, the coder's task is two-fold: he must first decide on the unit of the data to be coded, and then how to classify the unit. Boundaries of complex units are not self-evident; the coder must thus make a series of unitization decisions. This extra labor tends to lower intercoder agreement.

Unitization–deciding, for example, how many themes are in a stretch of data and where they begin or end–is a major hurdle. Studies have been reported in which most of the errors in thematic coding were attributable to unitization difficulties (Stempel, 1955).

Such errors can be reduced by coding only those complex units that can be linked to observable events in the data. Thus a complex behavioral orientation in classroom interaction would only be coded if it included a particular specific behavior. When coding complex units, it is also especially important that coders understand your explicitly stated categorization criteria and rules.

Paradoxically, you choose between simple and complex units by balancing the values of reliability and validity. A decision that increases reliability (a decision to code simple units, for example) may reduce the validity of the results, but without reliability there will be no validity at all.

III. CATEGORIZING THE DATA

In categorizing the data, it is essential to have a set of categories that can produce answers to your research questions while also being appropriate for the real data

with which you must work. The categories are the analytical concepts. They should be abstract enough to include all items (each item or unit being an operationalization of the concept). The closer the categories are to your theoretical purposes or hypotheses, the more valid will be your results. That is, the more likely it will be that your results explain the world the way you say they do. On the other hand, the rules of categorization have to be concrete enough and close enough to your data to permit competent and reliable coding.

Building the Category Set

Let's say you have a videotape record of a teacher's behavior over a sample of class meetings. How do you build a category set?

First, consider the hypotheses you are testing or the profiles you are attempting to construct. A list of categories should be exhaustive of these factors and should also be exhaustive of the data. Your preliminary list may include different teacher-asking behaviors, teacher-supervising behaviors, teacher-assigning behaviors and teacher-demonstrating behaviors.

Use the list to try to code a sample of the data. Perhaps some of the categories are seldom used and, upon reflection, don't appear to be essential. These categories should either be subsumed within others that are closely related, or dropped. (If such categories are dropped, the "other" category will serve in their place.) Similarly, when several categories seem to be measuring the same data attributes, perhaps they should be combined.

If many units in the data seem classifiable only under the residual "other" category, and if some of these hold something in common, then new categories may have to be added.

When you have finished, you may have combined several teacher-asking behaviors into one, you may have added categories to the set of teacher-assigning behaviors, and you may have eliminated several categories of teacher-supervising behaviors. If you have done the job well, categories within each set will be both mutually exclusive and exhaustive. To put this another way, there should be a category—and only one category—appropriate for each unit of data (Selltiz *et al.*, 1959). Your categories will now be ready for use in coding.

Relationship of Categories Within a Set

The relationship of categories within a set helps to determine your choice of a reliability statistic and affects the character of your data. If categories in a set are arranged along a continuum, they form a *scalar code.* If they are not so arranged, they form a *discrete-category code.*

Scalar Code. In the scalar code, categories differ only in the degree of magnitude of the variable being classified. For example, SRC (1965) used a 5-point code to classify answers to this question:

"In general, are you satisfied or dissatisfied with the way the United States has been acting towards other countries?"

(1) very satisfied
(2) satisfied
(3) both pro-and-con, or neutral
(4) dissatisfied
(5) very dissatisfied
(6) don't know, not ascertained other

Responses lying along the "satisfaction" continuum are grouped into the appropriate segments. The number of segments in the set depends on research needs and the coder's ability to make fine distinctions. A scalar code can always be collapsed into fewer points if it becomes necessary to raise intercoder agreement.

If the intervals separating each category along the continuum cannot be specified as being equal, then your code has an *ordinal* metric. If the intervals between the categories are equal, then your code has an *interval* metric. Later, when you calculate coding reliability, your choice of the appropriate correlational statistic will be based on this distinction.

Discrete-Category Code. When categories are not ordered along any continuum, they form a discrete-category code. The variety of categories is limited only by the number of questions one seeks to answer about his data and by the constraints discovered in the process of building the category sets. This is true whether the data pertain to love songs, political speeches, children's books, elements of classroom interaction or newspaper editorials. The metric of discrete-codes is *nominal;* that is, the coder decides whether a unit fits under one category or the other. The categories are different, but do not lie along any continuum. This attribute of discrete codes requires the use of agreement statistics in calculating coding

reliability, as will be discussed later.

Designing the Discrete-Category Code

How should the discrete-code be designed to be clear to coders? Should the categories be listed in a "menu format," forcing the coder to run his eyes down all of the list of alternatives before making his classification of an item? Or can they be ordered more effectively? Schutz (1958-59) listed some of the problems with the menu format:

(1) It is difficult for a coder to keep six or eight categories in mind at once. There are too many categories to keep in focus and as a result selection becomes subject to whim.

(2) When coders disagree about the classification of an item into a menu of categories, it is difficult to tell where in the decision process the coders diverged.

(3) Sometimes the categories only appear to be parallel when in fact one takes precedence over another and must be considered before the other in classifying a unit of the data.

Binary Method. Schutz (1958-59) developed the "binary method" to enable the researcher to provide a decision strategy for his coders. The binary method is comprised of a series of yes-no decisions by which all possible categories for each response are eventually exhausted. Larger categories or more general categories are transformed into series of smaller ones. Criteria are specified for each yes-no choice. The most general dichotomy is listed first, progressing down to the most specific categorical decision.

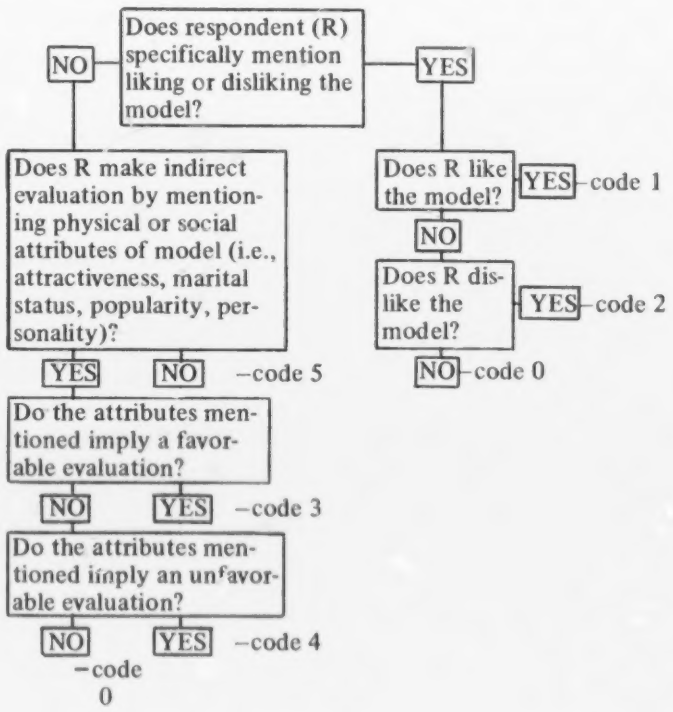
Schutz compared coding reliabilities produced by menu format and the binary method. He used the following categories to classify comic strip environments: United States, rural, historical, interstellar, urban and foreign. Arranged into a dichotomous series, the categories looked like this:

Decision 1 Interstellar ——— Earth
Decision 2 Foreign ———┘-United States
Decision 3 Historical ———└Contemporary
Decision 4 Rural ———└Urban

As you may have guessed, Schutz found that agreement among six coders improved measurably when the binary method was used rather than a menu format.

The psychological advantage of ordering coding decisions in a series of yes-no choices makes this form useful even when no case can be made that the first decision is more general than later ones.

Binary Code. Funkhouser (1966) used a binary code which resembled Schutz's in that lower order codes were subsumable into higher order ones. Suppose you were studying high school girls' perceptions of women in different occupational roles. Let's say you asked girls to describe their images of women doctors. You might use the following short binary code to categorize the respondents' liking of the woman doctor model:



(Extra decisions, and codes, could be added to take care of neutral or "both like and dislike" situations.)

Funkhouser noted that the binary code led to improved reliability with two previously troublesome category sets. It also led to data collapses that had not been obvious in earlier codes. In general, recasting the codes in binary form causes researchers to make a more rigorous interpretation of the data. They frequently structure decision points in terms of higher order concepts that had not been apparent in the original codes.

IV. CALCULATING RELIABILITY

Satisfactory intercoder agreement, or reliability, is the central criterion of success in coding. It demonstrates that a coder's work is based on a shared, rather than idiosyncratic, interpretation of the

categories. Reliability is measured by testing the equivalence of results when different coders classify the same data using the same set of categories.

Because reliability calculations are meant to serve as a diagnostic for the researcher, many researchers devise their own specific methods of computing reliability. The researcher wants to know whether his coders are agreeing enough (he decides how much is enough), whether his categories are ambiguous in places, and where. A low reliability coefficient, no matter how it is calculated, should cause the researcher to change either his coders or his code, whichever appears to be at fault. Viewed in this light, it is clear that the "significance" of the reliability coefficient is not at question–it matters only that reliability be high enough for the purposes of the research. A commonly required level of agreement is .85, but each researcher has to decide if he needs a higher or lower figure.

What the Statistic Should Do. To be most useful, a reliability statistic should do the following:

(1) It should take into account all the data that are available. It should control for attributes of the data that might affect the calculation.

(2) It should provide information about the sources of low reliability–whether the problem is in the code or in the coders.

(3) It should take into account the distribution of classifications used by each coder. How many categories there are and how much each is used will affect the percentage of agreement that is expected by chance, thus also affecting the meaning of the intercoder agreement figure that is obtained. In a category set comprised of two categories, one would expect by chance alone a 50% agreement between coders. If there were four categories in the set, then 25% agreement by chance would be expected. The higher the expected agreement by chance, then the higher the required obtained agreement should be.

(4) The statistic should be appropriate to the metric of the data. Statistics meant for nominal, ordinal or interval codes should be applied accordingly.

(5) In the case of ordinal and interval codes, the statistic should take into account "near-misses" in addition to simple disagreements. If one coder, on a 5-point scale, scores a statement as "5," his disagreement with another's score of "4" is one of degree and is less than his disagreement with another's score of "3."

Since no single reliability calculation fulfills all of these requirements, you may wish to use more than one statistic to assess your coding reliability.

The following section describes in greater detail the various reliability calculations. The reader who is not right now confronted with a need for these statistics may wish to turn to Section V, the conclusion.

Formulas

A. Correlation

When the category set is scalar, you should employ a correlational reliability calculation. Correlation takes into account degrees of disagreement, thus meeting the "near-miss" criterion.

(A scalar code is one in which something coded 2 has more of what is being considered than does something coded 1. To put it another way, a continuum exists.)

When two coders have produced *interval* data using a scalar code (where the intervals separating each category along the continuum are equal), the best reliability statistic is *Pearson's product-moment correlation coefficient (r).* You can find the formula for "r" in any standard statistics text. With the "r," you are correlating the decisions of two coders across the array of coded items. Low correlation between the coders due to differences between the scores they have given to the same units indicates problems in the code or in the coders.

When two coders have produced *ordinal* data using a scalar code (where the intervals separating categories along the continuum vary in size), a rank correlation coefficient such as *Spearman's rho* should be used to test reliability. The formula for "rho" can be found in standard texts on statistics (see Siegel, 1956). This statistic should also be used if the data produced by interval codes are severely skewed. As with the "r," with "rho" you are correlating the decisions of two coders across the array of coded items.

More Than Two Coders. If there are more than two coders, then you should calculate reliability with a statistic that includes the scores of all the coders. One way would be to average the correlations between all possible pairs of coders. This could become tedious: for example, 10 coders produce 45 pairs to be averaged. An easier way would be to use an N-coder or multi-coder correlational statistic such as the *Kendall Coefficient of Concordance W* which you can find in Siegel (1956). This coefficient computes the rank correlation of the codes across all pairs of coders (W is the linear function of the average *rho* between all possible pairs of rankings).

Weakness of Correlation. The weakness of correlational reliability coefficients is that they do not tell you whether the classifications by two or more coders are identical–only that they are proportional. Thus a perfect correlation of +1.0 is possible when coders never once agree, but where one is consistently a notch higher or lower than the other. This is illustrated in computing the correlation between coders 1 and 2, where coder 1 scores each unit in the data lower than coder 2 by a single point each time:

Items	Scores	
	Coder 1	*Coder 2*
Unit of data (A)	5	6
Unit of data (B)	1	2
Unit of data (C)	7	8
Unit of data (D)	3	4
Unit of data (E)	4	5

Using a standard formula to compute the correlation (r) between the two coders, it can be shown that the correlation equals +1.

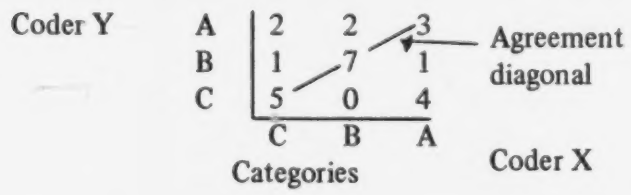
Thus correlation, which meets the "near-miss" criterion, can be deceptive. You still will want an indication of actual agreement between coders.

B. Agreement Statistics

Agreement statistics will be described in the context of nominal data produced by discrete-category codes. They can be used, however, with ordinal or interval data as checks on correlational coefficients.

In general, agreement reliability coefficients measure the proportion of actual agreements between coders over the total possible number of agreements. When it is computed using a cross-tabulation of coders' classifications, the simple *proportion of agreements* clearly has both face validity and diagnostic value.

Assume a pair of coders, coding 25 units into 3 (A, B, C) categories. The cross-tabulation table of their scores might be constructed like this:



Classifications in which coders agree lie on the agreement diagonal–5 agreements in cell (C, C), 7 in Cell (B, B), and 3 in cell (A, A). The other cells mark the frequencies of specific disagreements. For example, there are four units for which coder X scored an "A" while coder Y scored a "C." The proportion of agreement, the number of agreements (15) divided by the number of possible agreements (25) is .60, which may or may not be satisfactory, depending on the researcher's goals.

The matrix has diagnostic value because the researcher can see that category A gives more trouble than B, or C. There are nine occasions of error involving A, and there seems to be a particular difficulty distinguishing between A and C. The matrix, or cross-tabulation table, reveals any systematic disagreement between coders–with ordinal data, for example, it would show whether one coder was giving consistently higher (or lower) scores than another coder. A refinement of this kind of diagnostic is the *Random-Systematic-Error (RSE) coefficient,* which will be explained later.

While it does have face validity, the simple proportion of agreement statistic does not take into account the distribution of classifications in the category set. Nor does it correct for the fact that the fewer the categories there are in the set, the higher the chance that agreement will occur. That is, an .85 level of agreement will mean less when there are fewer categories in the set, and the researcher may be falsely confident that he has achieved satisfactory reliability.

Most Popular Agreement Statistic. Probably the most popular agreement statistic is *Scott's Pi* (Scott, 1955) which Holsti (1969) recommends. It accounts both for the number of categories and for the extent to which each is used. The formula is:

$$Pi = \frac{Po\ (\%\ \text{agreement}) - Pe\ (\text{expected}\ \%\ \text{agreement by chance})}{1 - Pe}$$

Pe is calculated by (a) collating the codings of both coders on a random set of responses that both have coded, (b) computing the proportions of scores which are placed by both of them together in each category of the set, (c) squaring the proportions, (d) and summing them.

For example, in the cross-tabulation table shown above, the proportions of scores in categories A, B, and C are .30, .36 and .34. Squared and summed, Pe = .34, which is the expected agreement if each coder had classified the responses randomly. The obtained proportion of agreement is .60. Then Pi, which may vary from 0.0 to 1.0 regardless of the number of categories,

equals:

$$\frac{.60 - .34}{1 - .34} = .39$$

Clearly, Scott's Pi is a more conservative statistic than the simple proportion of agreement. It is closest to the proportion of agreement when there are many categories, each used by coders with equal frequency, producing minimum agreement expectable by chance. Deviation from this rectangular distribution increases Pe, and thus reduces Pi.

Scott's Pi is particularly useful because it enables others to assess your reliability knowing that the number of categories and extent of their use have been taken into account. The Pi permits researchers to compare the coding reliabilities achieved in different studies.

Refined Diagnostic. Scott's Pi is an excellent summary statistic but does not tell the researcher very much about where the problems are if the Pi is low. Parker and Funkhouser (1968) developed the *Random-Systematic-Error (RSE) coefficient* to supplement Pi. The RSE shows whether a low reliability coefficient (preferably Scott's Pi) is due to coder error (lack of training, misunderstanding, frame of reference difficulty) or to ambiguous codes. Its diagnostic power is based on the logic that errors resulting from a defective code tend to scatter about the range of possible disagreements, while those errors resulting from coder error tend to occur in systematic patterns. In brief, the RSE tells whether and to what extent error patterns are systematic, and thus whether the problem is with the code or with the coders.

The following illustrative matrix might be constructed for two coders on the same set of responses, with 6 categories and 50 units to classify:

	Categories	Errors (Fy)						
	A	6	10	2	4*			
	B	0		5				
Coder Y	C	1	1		15			← Agreement
	D	1	1			4		Diagonal
	E	5	3		2			
	F	3	1			2		
			6	0	6	2	0	0 (Errors-Fx)
			A	B	C	D	E	F (Categories)

Coder X

RSE's are computed for the single cells in the matrix that do not fall on the agreement diagonal, and for the marginal cells labelled Fy and Fx. The single cells show the number of times that units were classified as belonging to one category by one coder, and belonging to another category by the other coder. Thus the cell (A, C), marked with an asterisk in the matrix, shows that there were four errors in which Coder Y classified a unit as belonging to category A, while Coder X classified it as belonging to Category C.

The marginal cells, Fx and Fy, are the sums of the errors added row-wise and column-wise.

Parker and Funkhouser (1968) provide the formula to compute the RSE's for single cells and marginal cells. It suffices to say here, to demonstrate the logic of this diagnostic, that:

(1) if the RSE in single cells is above a certain point determined in the computation–the demarcation point–then disagreements are systematically falling into single cells, which may indicate coders' disagreement over a particular type of response;

(2) if either marginal RSE is above the demarcation point, then disagreements are systematically falling into certain categories with respect to one of the coders–this may be the result of misinterpretation or misuse of the code;

(3) if both marginal RSE's are high, it may be the result of both coders making systematic errors, or it may be an artifact of high RSE's in single cells;

(4) when RSE in single cells and the two marginal RSE's all fall below their demarcation points, then the fault for a low reliability coefficient probably lies in defective code.

All the agreement statistics discussed thus far are designed for two-coder situations. If there are more than two coders, the statistics have to be modified. One way you could do this would be to compute the Pi for all possible pairs of coders and then average the reliability estimates across these pairs. However, to avoid the tediousness of this exercise, it is better to apply N-coder agreement statistics when there are more than two coders.

N-Coder Agreement Statistic. Stempel (1955) developed a calculation that indicates the degree to which any single coder is in agreement with the majority, and that estimates overall reliability by averaging the coders' agreement scores. Each coder's agreement score equals the percentage of responses where the coder agreed with the majority. Where there is

no majority on an item, then each coder is given a "no agreement" mark.

Stempel's computation has diagnostic value because it points out the performances of individual coders against a common criterion. It also highlights problem categories, which can be identified in the matrix of items by coders.

V. CONCLUSION

In this brief and simple guide to coding, you have read about the place and purpose of coding in the research process, about the trade-offs involved in unitizing your data, about building and designing category sets, and about the theory and practice of calculating coding reliability.

Clearly, there are pitfalls and complexities, but these should not discourage you. Rather you should consider yourself fore-warned and thus fore-armed.

Many educational and social researchers are now working in areas where data are extremely rich but cannot be pre-structured for collection and analysis.

Thus coding, often considered a mundane task, becomes an essential research tool. The researcher who chooses well at each decision-point in coding is the one who will get the most out of his data.

REFERENCES

B. Frisbie and S. Sudman, "The Use of Computers in Coding Free Responses," *Public Opinion Quarterly*, Vol. 32, Issue 2, Summer 1968, pp. 216-232.

A comparison of human and computer coding of open-ended questionnaire responses, where they were found to be about equal, and a further test of computer coding of underlying themes, plus discussion on computer's performance.

G. R. Funkhouser, "Appendix VI, Binary Coding," in E. B. Parker and W. J. Paisley, eds., *Patterns of Adult Information Seeking*, Institute for Communication Research, Stanford University, 1966, 275 pp. Available from the ERIC Document Reproduction Service, P.O. Drawer O, Bethesda, Maryland 20014 in microfiche for 65 cents and hardcopy for $9.87 as document ED 010 294.

Provides rationale for "binary code" ordering of categories in a set, reports tests of this method which showed improved reliability, and gives examples.

G. R. Funkhouser and E. B. Parker, "Analyzing Coding Reliability: The Random-Systematic-Error Coefficient," *Public Opinion Quarterly*, Vol. 32, Issue 1, 1968, pp. 122-128.

A technical discussion on sources of coding error with description of the authors' diagnostic statistic, the Random-Systematic-Error coefficient, and instructions for its use.

G. Gerbner, O. R. Holsti, K. Krippendorff, W. J. Paisley and P. J. Stone, eds., *The Analysis of Communication Content, Developments in Scientific Theories and Computer Techniques*, John Wiley & Sons, Inc., New York, 1969, 597 pp., $14.95.

An excellent collection of 29 papers on theory and practice of content analysis that includes state-of-the-art discussions on aspects of inference from content data, recording and notation of data, and computer techniques.

D. H. Goldhamer, "Toward a More General Inquirer: Convergence of Structure and Context of Meaning," in G. Gerbner, O. R. Holsti, K. Krippendorff, W. J. Paisley and P. J. Stone, eds., *The Analysis of Communication Content, Developments in Scientific Theories and Computer Techniques*, John Wiley & Sons, New York, 1969, pp. 343-353, $14.95.

Technical discussion of General Inquirer procedures and overview of major linguistic issues facing computerized coding of verbal material.

O. R. Holsti, *Content Analysis for the Social Sciences and Humanities*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1969, 235 pp., $4.75.

The classic basic text on content analysis. Thorough discussions on all facets of coding.

W. C. Schutz, "On Categorizing Qualitative Data in Content Analysis," *Public Opinion Quarterly*, Vol. 22, Issue 4, 1958-1959, pp. 503-515.

A useful technical guide to building category sets, including analysis of and comparison between different methods of ordering categories within a set.

W. Scott, "Reliability of Content Analysis: The Case of Nominal Scale Coding," *Public Opinion Quarterly*, Vol. 19, Issue 3, 1955, pp. 321-325.

Useful technical discussion on methods of computing coding reliability. Provides rationale for use of *Pi* as an index of inter-coder agreement.

C. Selltiz, M. Jahoda, M. Deutsch and S. W. Cook, *Research Methods in Social Relations* (Revised), Holt, Rinehart and Winston, New York, 1959, 622 pp., $11.50.

A basic text on social research methods, including a short section on coding.

S. Siegel, *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill Book Company, Inc., New York, 1956, 312 pp., $10.50.

A very useful compendium of statistical tests that do not require normally-distributed data, including the various rank correlations.

G. H. Stempel III, "Increasing Reliability in Content Analysis," *Journalism Quarterly*, Vol. 32, Issue 4, 1955, pp. 449-455.

Explores sources of error in coding simple and complex units. Also introduces an N-coder reliability calculation that is based on consensus of coding decisions.

P. J. Stone, "Improved Quality of Content-Analysis Categories: Computerized-Disambiguation Rules for High Frequency

English Words," in G. Gerbner, O. R. Holsti, K. Krippendorff, W. J. Paisley and P. J. Stone, eds., *The Analysis of Communication Content, Developments in Scientific Theories and Computer Techniques,* John Wiley & Sons, Inc., New York, 1969, pp. 199-221, $14.95.

Describes computer procedures to help computer coder recognize the sense or meaning of a word based on its context of preceding and following words.

Survey Research Center, *A Manual for Coders,* Institute for Social Research, University of Michigan, Ann Arbor, 1955, 58 pp.

A detailed basic guide to all facets of sample surveying and of coding questionnaire responses, including description of the SRC's own procedures.

This paper is distributed pursuant to a contract with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under government sponsorship are encouraged to express freely their judgment in professional and technical matters. Points of view or opinions do not, therefore, necessarily represent official Office of Education position or policy.